

AD-A159 029

PARALLEL IMAGE PROCESSING AND IMAGE UNDERSTANDING(U)
MARYLAND UNIV COLLEGE PARK CENTER FOR AUTOMATION
RESEARCH A ROSENFELD 05 JUL 85 AFOSR-RR-85-0645

1/1

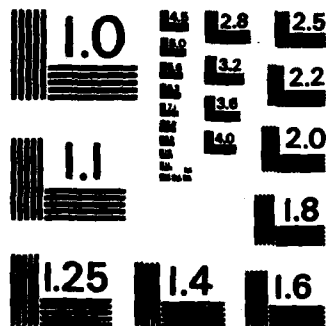
UNCLASSIFIED

F49620-83-C-0082

F/G 9/2

NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED		1b. RESTRICTIVE MARKINGS	
2a. SECURITY CLASSIFICATION AUTHORITY		3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited.	
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE		5. MONITORING ORGANIZATION REPORT NUMBER(S) AFOSR-TR- 85-0645	
4. PERFORMING ORGANIZATION REPORT NUMBER(S)		7a. NAME OF MONITORING ORGANIZATION Air Force Office of Scientific Research	
6a. NAME OF PERFORMING ORGANIZATION University of Maryland	6d. OFFICE SYMBOL (If applicable)	7b. ADDRESS (City, State and ZIP Code) Directorate of Mathematical & Information Sciences, Bolling AFB DC 20332	
6c. ADDRESS (City, State and ZIP Code) College Park, MD 20742	8a. NAME OF FUNDING/SPONSORING ORGANIZATION AFOSR	8b. OFFICE SYMBOL (If applicable) NM	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER F49620-83-C-0082
8c. ADDRESS (City, State and ZIP Code) Bolling AFB DC 20332	10. SOURCE OF FUNDING NOS.		
	PROGRAM ELEMENT NO. 61102F	PROJECT NO. 2304	TASK NO. A7 42
11. TITLE (Include Security Classification) Parallel Image Processing and Image Understanding			
12. PERSONAL AUTHOR(S) Azriel Rosenfeld			
13a. TYPE OF REPORT Final	13b. TIME COVERED FROM 01 Sep84 TO 1 Apr85	14. DATE OF REPORT (Yr., Mo., Day) 5 July 1985	15. PAGE COUNT 8
16. SUPPLEMENTARY NOTATION			
17. COSATI CODES		18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)	
FIELD	GROUP	SUB GR.	
XXXXXXXXXXXXXX		parallel algorithms, processing images, interprocessor communication	
19. ABSTRACT (Continue on reverse if necessary and identify by block number)			
<p>20 technical reports were issued under the contract during this period. Most of these reports have resulted in, or are expected to result in, published papers. Abstracts of the reports are given below in chronological order. (Numbers in brackets refer to these abstracts.) The reports deal with the following topics: a) Parallel algorithms and architectures for processing images (1,6,17,18,19) and data structures derived from images (2, 14, 16) b) Interprocessor communication for parallel processing (5,10,11,15) c) Knowledge-based image understanding (4,13) d) Image modeling (texture shape) (7,8,9,12,20). Research is continuing in all of these areas under a successor contract.</p>			
20. DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED <input checked="" type="checkbox"/> SAME AS RPT. <input type="checkbox"/> DTIC USERS <input type="checkbox"/>		21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED E	
22a. NAME OF RESPONSIBLE INDIVIDUAL Robert N. Buchal	22b. TELEPHONE NUMBER (Include Area Code) (202) 767-4939	22c. OFFICE SYMBOL NM	

DTIC FILE COPY

**DTIC
ELECTE
SEP 11 1985**

AFOSR-TR- 85 - 0645

**FINAL REPORT
on Contract F49620-83C-0082**

**for research on
Parallel Image Processing and Image Understanding**

**Submitted to: U.S. Air Force Office of
Scientific Research
Bolling Air Force Base
Washington, DC 20332**

**Submitted by: Center for Automation Research
University of Maryland
College Park, MD 20742**

Principal Investigator: Azriel Rosenfeld

July 5, 1985

**Approved for public release;
distribution unlimited.**

Accession For	
NTIS GRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input checked="" type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A-1	



**AIR FORCE OFFICE OF SCIENTIFIC RESEARCH (AFSC)
NOTICE OF THIS REPORT TO DTIC
This report has been reviewed and is
approved for distribution under AFOSR-12.
MATTHEW J. K...
Chief, Technical Information Division**

85 9 09 085

(1)

This report summarizes the research conducted under Contract F49620-82-C-0082 during the period May 1983 - April 1985. 20 technical reports were issued under the contract during that period. Most of these reports have resulted in, or are expected to result in, published papers. Abstracts of ²⁰the reports are given below in chronological order. (Numbers in brackets refer to these abstracts.) *This bibliography* The reports deal with the following topics:

- a) Parallel algorithms and architectures for processing images [1,6,17,18,19] and data structures derived from images [2,14,16] (see also [3] on image processing software)
- b) Interprocessor communication for parallel processing [5,10,11,15]
- c) Knowledge-based image understanding [4,13]
- and* d) Image modeling, (texture, shape) [7,8,9,12,20]

Research is continuing in all of these areas under a successor contract.

ABSTRACTS OF TECHNICAL REPORTS

1. Roger Eastman, "Reliability in Cellular Arrays Through Data Sharing", CS-TR-1285, CAR-TR-7, May 1983.

This paper describes several methods for increasing cellular array reliability for image processing through redundancy, with emphasis on error masking methods that preserve continuous operation of the array. We define n -module data sharing (NMDS) for a cellular array, which achieves redundancy by sharing data and computations between adjacent processors in the array. The time, space and reliability performance of NMDS is compared to standard n -module redundancy (NMR).

2. Simon Kasif and Reinhard Klette, "A Data Allocation Problem for SIMD Systems", CS-TR-1292, CAR-TR-11, June 1983.

Let $S = \{S_1, S_2, \dots, S_N\}$ be a set of data objects, and f be function from $S \times S$ into S . For $T \subseteq \{1, 2, \dots, N\}^2$ the operation of T_f applies f to every element of $\{(S_i, S_j) \mid (i, j) \in T\} \subseteq S \times S$. For the optimal computation of T_f on a SIMD system with p processors, the data allocation problem consists in distributing objects to the p processors such that the computation and data transfer time is minimized. In the paper, the cases $T = \{1, 2, \dots, N\}^2$ and $T = \{(i, j) \mid 1 \leq j < i \leq N\}$ are dealt with.

3. Reinhard Klette, Klaus Voss, and Peter Hufnagl, "Interactive Software Systems for Computer Vision", CS-TR-1293, CAR-TR-12, June 1983.

This paper gives an overview of the main directions of program system development in computer vision, where special emphasis is paid to interactivity. A systematic representation of interactive computer vision systems is given. Brief information about 39 different software systems is listed in an Appendix.

4. Vincent Shang-Shouq Hwang, Takashi Matsuyama, Larry S. Davis, and Asriel Rosenfeld, "Evidence Accumulation for Spatial Reasoning in Aerial Image Understanding", CS-TR-1336, CAR-TR-28, October 1983.

We describe a control structure for building an Image Understanding System. This system can deal with objects with diverse appearances when consistent spatial relations exist between objects. By accumulating consistent predictions originated by existing instances, our system can dynamically reason about what to do in order to construct interpretations of the image. In this paper, we have discussed parts of the proposed system — the representation of spatial knowledge, the accumulation of evidence, the focus of attention mechanism, and the integration of constraints for top-down control.

5. I.V. Ramakrishnan, D.S. Fussell, and A. Silberschatz, "On Mapping Homogeneous Graphs on A Linear Array-Processor Model", CS-TR-1339, CAR-TR-30, October 1983.

This paper presents a formal model of linear array processors suitable for VLSI implementation as well as graph representation of programs suitable for execution on such a model. A distinction is made between correct mapping and correct execution of such graphs on this model and the structure of correctly mappable graphs is examined. The formalism developed is used to synthesize algorithms for this model.

6. I.V. Ramakrishnan, "Modular Matrix Multiplication On A Linear Array" CS-TR-1340, CAR-TR-31, November 1983.

A matrix-multiplication algorithm on a linear array using an optimal number of processing elements is proposed. The local storage required by the processing elements and the I/O bandwidth required to drive the array are both constants that are independent of the sizes of the matrices being multiplied. The algorithm is therefore modular, that is, arbitrarily large matrices can be multiplied on a large array built by cascading small arrays. The array is well-suited for VLSI implementation.

7. Songde Ma and André Gagalowicz, "A Parallel Method for Natural Texture Synthesis", CS-TR-1343, CAR-TR-32, November 1983.

This paper deals with an optimization technique applied to natural texture synthesis. We propose a definition of a global criterion which is the mean square error between the statistical features of a natural original texture and those of an artificially generated one. A gradient algorithm is used to minimize this criterion. The statistical feature vector used was the autocorrelation function although this is by no means the only choice. The textures generated are very similar to the original ones. This method can be implemented in a highly parallel manner.

8. André Gagalowicz and Song De Ma, "Synthesis of Natural Textures on 3-D Surfaces", CS-TR-1344, CAR-TR-33, November 1983.

This paper presents a new method for the synthesis of textures on 3-D surfaces. To our knowledge, one basic technique has been presented up to now in the literature (see [7-18]). In this standard method, textures are synthesized by mapping a rectangular template onto the curved surface. This method is complex, requires substantial computing time, and presents some drawbacks such as the possibility of obtaining aliasing effects and continuity problems along the edges of the curved templates. Procedures to eliminate these problems are available [11,12] but make this synthesis even more unattractive. The method proposed in this paper does not present the former drawbacks. We do not use a template

mapping, which is a drawback in itself. The synthesis is achieved continuously on the surface, so that there are no edge effects and also no aliasing effects. This method is a simple extension of a procedure that we have proposed before in the literature [2,3] for planar textures. Any kind of texture can be reproduced with a good similarity to the reference texture used. It also has the important advantage that only one set of second order statistics (a small amount of data) needs to be computed on a planar version of the reference texture to synthesize this texture on any surface and at any distance. Some results on simple surfaces are displayed (cylinder, sphere), but the method holds for any surface and is relatively quick and easy.

9. André Gagalowicz and Song De Ma, "Sequential Synthesis of Natural Textures", CS-TR-1345, CAR-TR-34, November 1983.

A new method for the generation of natural textures is presented. Using a priori given second order statistics (second order spatial averages or autocorrelation parameters) of a natural texture as input, we give a procedure to synthesize an artificial texture field in such a way that its second order statistics are equal to the desired ones. The synthesis is achieved directly without inventing higher order statistics, as was the case in earlier publications [3-7,15]. This method allows us to synthesize gray tone texture fields while "controlling" their second order statistics in rather large neighborhoods. The synthesized textures are very similar visually to the original natural textures used to compute the second order statistics, but second order spatial averages give better results than autocorrelation parameters. This seems to strongly support the conjecture [6,8,10] that the visual system is only sensitive to the second order spatial averages of a given texture field, so that these statistics should be used to model textures.

10. Angela Y. Wu, "Embedding of Networks of Processors Into Hypercubes", CS-TR-1354, CAR-TR-36, December 1983.

The hypercube is a good host graph for the embedding of networks of processors because of its low degree and low diameter. Graphs such as trees and arrays can be embedded into a hypercube with small dilation and expansion costs, but there are classes of graphs which can be embedded into a hypercube only with large expansion cost or large dilation cost.

11. LV. Ramakrishnan and P.J. Varman, "On Mapping Cube Graphs on VLSI Array and Tree Architectures", CS-TR-1358, CAR-TR-41, December 1983.

We formalize a model of array architectures suitable for VLSI implementation. A formal model of an arbitrarily structured tree machine is also presented. A mathematical framework is developed to transform cube graphs, which are data-flow descriptions of certain matrix computations, onto the array and tree models. All published algorithms for these computations can be obtained using the

mathematical framework. In addition, novel linear-array algorithms for matrix multiplication are obtained. More importantly, the algorithms obtained for the tree model are of special significance. Besides their novelty, the independence of the tree algorithms from a specific inter-processor communication geometry make them robust to hardware faults as opposed to algorithms that are based on specific interconnection requirements.

12. Azriel Rosenfeld and Reinhard Klette, "Degree of Adjacency or Surroundedness", CS-TR-1380, CAR-TR-53, March 1984.

Definitions of the degree of adjacency of two regions in the plane, and the degree of surroundedness of one region by another, are proposed. Some elementary properties of these concepts are established, and it is also shown that they have natural generalizations to fuzzy subsets of the plane. Applications of the proposed measures to digital polygons are demonstrated and fast algorithms for computing these measures are given.

13. Takashi Matsuyama, Vincent Shang-Shouq Hwang, and Larry S. Davis, "Evidence Accumulation for Spatial Reasoning", CS-TR-1381, CAR-TR-54, March 1984.

This paper describes the evidence accumulation process of an image understanding system first described in [1], which enables the system to perform top-down (goal-oriented) picture processing as well as bottom-up verification of consistent spatial relations among objects.

14. I.V. Ramakrishnan, "Dynamic Programming and Transitive Closure on Linear Pipelines", CS-TR-1388, CAR-TR-57, May 1984.

Algorithms for the dynamic programming and transitive closure problems are presented for a linear pipeline of processors. These algorithms require only a constant number of I/O ports and are optimal in their area and time requirements.

15. S.K. Bhaskar, Azriel Rosenfeld and Angela Y. Wu, "Simulation of Large Networks of Processors By Smaller Ones", CS-TR-1401, CAR-TR-63, May 1984.

This paper considers the problem of simulating a large network N of processors using a small set of p processors. The approach taken is to partition the nodes of N into p subsets N_1, \dots, N_p and to assign each subset to a processor for simulation. In order to equalize the workloads of the processors, the sizes of N_1, \dots, N_p should be as equal as possible; and in order to minimize (and equalize) the amount of message passing between the processors, the number of pairs of nodes that are neighbors in N but belong to different subsets should be as small (and as equal) as possible. We discuss the general problem of

partitioning a graph N so as to satisfy these criteria, and also consider the particular case of partitioning a tree.

16. Simon Kasif, "Parallel Searching and Merging On ZMOB", CS-TR-1405, CAR-TR-64, June 1984.

One of the most difficult issues that must be addressed when studying a class of parallel algorithms is the problem of choosing a model that captures the inherent difficulty of implementing these algorithms on a multiprocessor architecture. Shared memory models have proven to be an effective tool for deriving lower bounds on the complexity of comparison problems. In particular, a speed-up of $\lg(P)$ is possible for the problem of finding an element in an N -element sorted list, and speed-ups of $P/\lg P$ and P are possible for merging N -element sorted lists on P processors for the cases of $N = P$ and $P < N$ respectively.

In practice, these speed-ups are not attainable since the shared memory models ignore many practical considerations in multiprocessor systems, such as interprocessor communications, distribution of data on local memories and limited fan-out of memory locations. In this paper we introduce a model for parallel computation that is strictly weaker than the shared memory models. The model is based on an actual machine currently being constructed (ZMOB). We examine the communication facilities available in the model and show that lower bounds for merging and searching on shared memory models are attainable (within a constant). The main results reported in the paper are:

- an $O(\lg N / \lg P)$ algorithm for searching an N -element sorted list distributed on P processors.
- an $O(N/P)$ algorithm for merging two N -element lists on $2P$ processors.
- an $O(\lg N)$ algorithm for merging two N -element lists on $2N$ processors.
- criteria and techniques for simulating CREW PRAM algorithms on ZMOB.

One of the techniques is used to establish an $O(\lg \lg N)$ lower bound for merging two N -element lists on $2N$ processors.

17. Azriel Rosenfeld, "The Prism Machine: An Alternative to the Pyramid", CS-TR-1416, CAR-TR-70, July 1984.

The prism machine is a stack of n cellular arrays, each of size $2^n \times 2^n$. Cell (i, j) on level k is connected to cells (i, j) , $(i + 2^k, j)$, and $(i, j + 2^k)$ on level $k + 1$, $1 \leq k < n$, where the sums are modulo 2^n . Such a machine can perform various operations (e.g., "Gaussian" convolutions or least-squares polynomial fits) on image neighborhoods of power-of-2 sizes in every position in $O(n)$ time, unlike a pyramid machine which can do this only in sampled positions. It can also compute the discrete Fourier transform in $O(n)$ time. It consists of $n \cdot 4^n$ cells, while a pyramid consists of fewer than $4^n + 1/3$ cells, but in practice n would be at most 10, so that a prism would be at most about seven times as large as a pyramid.

18. I.V. Ramakrishnan and Shaunak Pawagi, "Parallel Update of Minimum Spanning Trees in Logarithmic Time", CS-TR-1452, CAR-TR-97, November 1984.

Parallel algorithms are presented for updating a minimum spanning tree when the cost of an edge changes or when a new node is inserted in the underlying graph. The machine model used is a parallel random access machine which allows simultaneous reads but prohibits simultaneous writes into the same memory location. The algorithms described in this paper for updating a minimum spanning tree require $O(\log n)$ time and $O(n^2)$ processors. These algorithms are efficient when compared to previously known algorithms for initial construction of a minimum spanning tree that require $O(\log^2 n)$ time and use $O(n^2)$ processors.

19. Angela Y. Wu, "Parallel Processing of Encoded Bit Strings", CS-TR-1455, CAR-TR-98, November 1984.

Many operations on strings of length n can be speeded up by a factor of p using p processors. String operations can also be speeded up, even when a single processor is used, by compactly encoding the strings, e.g., using run length code. This paper shows how to combine these two approaches by using p processors to process compactly encoded strings.

20. Azriel Rosenfeld, "Axial Representations of Shape", CS-TR-1462, CAR-TR-102, December 1984.

Classes of "ribbonlike" planar shapes can be defined by specifying an arc, called the spine or axis, and a geometric figure such as a disk or line segment, called the generator, that "sweeps out" the shape by moving along the spine, changing size as it moves. Shape descriptions of this type have been considered by Blum, Brooks, Brady, and others. This paper considers such descriptions from the standpoints of both generation and recovery (i.e. given a shape generated in this way, to determine the axis and generation rule that gave rise to it), and discusses their relative advantages and disadvantages.

END

FILMED

10-85

DTIC